N88-25738

AN OPERATIONAL OPEN-END FILE TRANSFER PROTOCOL FOR MOBILE SATELLITE COMMUNICATIONS

Charles Wang, Unjeng Cheng and Tsun-Yee Yan
Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Drive
Pasadena, CA 91109

ABSTRACT

This paper describes an operational open-end file transfer protocol which includes the connecting procedure, data transfer, and relinquishment procedure for mobile satellite communications. The protocol makes use of the frame level and packet level formats of the X.25 standard for the data link layer and network layer, respectively. The structure of a testbed for experimental simulation of this protocol over a mobile fading channel is also introduced.

1. INTRODUCTION

The concept of mobile communications through a geosynchronous satellite has been studied extensively in recent years [1,2]. Currently, the Mobile Satellite Experiment (MSAT-X) project at the Jet Propulsion Laboratory (JPL) is aimed at validating the ground segment technologies for future generations of the Land-Mobile Satellite System (LMSS) [3,4]. Such a satellite-based mobile communications network can provide open-end and closed-end services to mobile subscribers throughout a vast geographical area. In [5], a generic concept of open-end and closed-end services has been presented.

In data communications, a set of rules, called a protocol, must be followed to set up and take down a logical link and ensure reliable transmissions between the sender and the receiver. To decouple the interactions between various functions of a protocol, most network protocols are organized as a series of layers. Each layer only interfaces with the layers one level above and below it. The purpose of this hierarchy is to shield each layer from the details of how other layers are actually implemented. Hence, the implementation of each layer can be modified and changed without affecting the implementation of other layers. The International Standards Organization (ISO) has developed an Open Systems Interconnection (OSI) Reference Model to standardize the protocol development. This reference model includes 7 layers, namely, from the lowest to the highest, physical layer, data link layer, network layer, transport layer, session layer, presentation layer, and application layer [6].

Developing a protocol for a mobile communications network is a challenging problem. Various protocols such as BISYNC, SDLC, ADCCP, HDLC, DDCMP, X.21, X.25, X.28, etc., have been developed for different applications. X.25 is the protocol which has been widely used to interface Data Terminal Equipment (DTE) with the packet switched data network. Specifically it comprises the lowest three layers of the ISO-OSI reference model. In this paper, we make use of the data link layer and network layer formats of X.25 for implementing the open-end file transfer service for mobile satellite communications. This allows an existing commercial DTE to access the mobile satellite network with minor modifications.

In this paper, Section 2 describes the conceptual open-end file transfer for mobile satellite communications. In Sections 3 and 4, we illustrate how X.25 format can be used for the network layer and data link layer, respectively. Finally, Section 5 presents the structure of a testbed for experimental simulation of this protocol in a mobile fading environment.

2. FILE TRANSFER FOR OPEN-END CONNECTIONS

In a mobile communications network, the service of transferring long data files is often necessary. Since the duration of a data file may not be known in advance, the network must maintain the allocation of the resource to the file transfer until termination. As stated in [5], a subscriber must send a connection request for transmission through a request channel to the Network Management Center (NMC) whenever it wishes to initiate an open-end connection. After sending out a request, the subscriber waits for an assignment packet, which also serves as an acknowledgment to the request, from the NMC. If the subscriber does not receive the assignment packet within a preset timeout period, it retransmits the request after a backoff delay.

Upon receiving a successful open-end connection request, the NMC checks whether the destination party is busy or whether all open-end channels are occupied. In either case, the NMC sends a busy status to the requester. Otherwise, the NMC attempts to notify the destination party. If this is successful, the NMC sends an assignment packet to the requester. If this assignment packet is received correctly, the requester tunes to the assigned open-end channel and starts transmitting the file. At the end of file transfer, the requester sends a relinquishment message to NMC for taking down the channel.

3. CONNECTING AND RELINQUISHMENT PROCEDURES USING X.25 PACKET LEVEL PROTOCOL

The connecting and relinquishment procedures described above can be implemented using the packet level logical interface of X.25 [6,7]. Figure 1 summarizes the control packets of X.25 used in our scenario. The procedures to establish and terminate an open-end connection are described in Figures 2 and 3, respectively. The frame structures of various control packets are illustrated in Figures 4(a) through 4(d). Note that the CALL-REQUEST and INCOMING-CALL packets must contain the addresses of the calling DTE and the called DTE. Furthermore, for INCOMING-CALL and CALL-CONNECTED packets, the channel # field is used to identify the assigned data-transmission channel.

To establish a file transfer session, the calling DTE sends a CALL-REQUEST packet to the NMC through a request channel. It then waits for a CALL-CONNECTED packet from the NMC. If the calling DTE does not receive the CALL-CONNECTED packet within T_{CR} seconds from the NMC, it will retransmit the same CALL-REQUEST packet.

After receiving the CALL-REQUEST packet successfully from the calling DTE, NMC sends an INCOMING-CALL packet to the called DTE if the called DTE is not busy and an open-end channel is available (referred to as the line-not-busy case, see Figure 2(a)); otherwise, it sends a CLEAR-INDICATION packet back to the calling DTE (referred to as the line-busy case, see Figure 2(b)). Note that the CLEAR-INDICATION packet is also used in the call-clearing procedure (see the following discussion). The INCOMING-CALL packet contains the identity of the assigned data-transmission channel and the scheduled transmission-start-time. The assigned channel is allocated by NMC at the moment that the INCOMING-CALL packet is issued. After receiving the INCOMING-CALL packet successfully, the called DTE responds with a CALL-ACCEPTED packet to NMC through the assigned data-transmission channel. Since the CALL-ACCEPTED packet is sent through the allocated channel, the address of the called DTE is not required. The CALL-ACCEPTED packet can be viewed as an acknowledgement to the INCOMING-CALL packet. In case the CALL-ACCEPTED packet is not received within a time-out period (T_{IC} seconds), NMC will retransmit the same INCOMING-CALL packet. Since the channel error rate is high in the mobile environments, the same INCOMING-CALL packet may be transmitted to the called DTE more than once. Hence the scheduled transmission-start-time on each retransmitted INCOMING-CALL packet may be different to account for the time delay caused by retransmissions. Furthermore, the time-out period for retransmitting the INCOMING-CALL packets must be long enough to ensure that the CALL-ACCEPTED packet received by NMC acknowledges the most recent INCOMING-CALL packet.

After successfully receiving the CALL-ACCEPTED packet from the called DTE, NMC sends a CALL-CONNECTED packet to the calling DTE. In case the CALL-CONNECTED packet is lost or not received correctly, the calling DTE will retransmit the CALL-REQUEST packet after a time-out period T_{CR}. The CALL-CONNECTED packet contains the identity of the assigned data-transmission channel and the transmission-start-time. After the calling DTE successfully receives the CALL-CONNECTED packet, the file transfer session will begin at the scheduled transmission-start-time.

During data transmission phase, NMC monitors the allocated channel for a CLEAR-REQUEST packet. The relinquishment procedure will be initiated by the calling DTE to send a CLEAR-REQUEST packet to NMC through the data-transmission channel. As shown in Figure 3, the relinquishment procedure is the dual of the line-not-busy case of the connecting procedure with the CALL-REQUEST, INCOMING-CALL, CALL-ACCEPTED and CALL-CONNECTED packets being replaced by the CLEAR-REQUEST, CLEAR-INDICATION, DTE-CLEAR-CONFIRMATION, and DCE-CLEAR-CONFIRMATION packets, respectively. Since NMC knows who are using the allocated channels, the terminal identity is not necessary in the CLEAR-REQUEST and DTE-CLEAR-CONFIRMATION packets.

4. DATA LINK CONTROL USING X.25 LINK LEVEL PROTOCOL

In the mobile environment, to ensure reliable communications between the sender and the receiver, an automatic retransmission protocol must be utilized. The data link level protocol, i.e. Link Access Protocol, Balanced (LAP-B), of X.25 provides an efficient retransmission scheme that can be used for the data link control in open-end file transfers.

Figure 5 shows the standard X.25 frame structure. In X.25, a flag pattern, 01111110, is used to indicate the beginning and the end of a frame. If this flag pattern is corrupted by the channel noise or fading, the entire content of the frame will be considered lost. Thus, in our applications, a new flag pattern is designed to cope with the high bit error rate in the mobile environment. The address field (ADDR) identifies the terminal to which the frame is sent. The control field (CTRL) is used for sequence numbers, acknowledgments, and other purposes, as discussed below. The data field (DATA) contains the data which is in the packet format from the network layer. In X.25, the DATA field may be arbitrarily long; however, our previous study showed that the DATA field length should be kept below 512 bits in order to achieve an acceptable packet error rate in the mobile fading environments [8]. The FCS field is a minor variation of the well-known cyclic redundancy code (CRC) which allows lost flag bytes to be detected.

There are three kinds of frames: Information (I), Supervisory (S), and Unnumbered (U), specified by the CTRL field. Figure 6 shows various formats of these frame types. The Information (I) frames encapsulate the data packets of the file and the control packets for the connecting and relinquishment procedures. The sequence numbers, N(S) and N(R), use 3 bits for which up to 7 unacknowledged frames may be outstanding at any instant. N(S) is the sequence number of the currently transmitted frame. N(R) is the expected sequence number of the next received frame.

In the Supervisory frames, Receiver-Ready (RR), Receiver-Not-Ready (RNR), and Reject (REJ) frames represent various kinds of acknowledgments. RR is an acknowledgment frame used to indicate the next expected frame. RNR acknowledges all frames up to but not including N(R). Although it looks like RR, it tells the sender to stop sending. This is to indicate certain temporary problems with the receiver, such as out of buffer space. When the problem is cured, the receiver sends RR, REJ or other control frames. REJ is a negative acknowledgment frame to the N(R) frame. The sender then must re-send all outstanding frames starting at N(R) in the current sliding window. This constitutes the Go-Back-N retransmission scheme. The X.25 does not provide a selective-repeat retransmission scheme. However, another similar data link protocol, High-Level Data Link Control (HDLC), does provide the selective repeat scheme by adopting an additional control field format, selective reject (SREJ). In that case, SREJ is also a negative acknowledgment frame to the N(R) frame. Nevertheless, it only asks for retransmission of the N(R) frame in sender's current sliding widow.

The unnumbered formats, SABM, UA, FRMR, and DISC, are used to report the status of the terminal or some odd situations of the transmission. They are not of the most importance as far as the normal file transfer is concerned and are not discussed here (see [9] for details).

5. A TEST-BED SIMULATOR FOR THE PROTOCOL

A test-bed which can experimentally simulate the proposed open-end file transfer protocol is developed. Figure 7 illustrates the configuration of this test-bed simulator. It consists of three major components, two programmable protocol analyzers and an IBM PC AT equipped with a 68020 co-processor board. The protocol analyzers perform the data link layer and network layer protocols and are used to emulate the network elements such as the calling party, the called party, and the NMC. The co-processor board inside the IBM PC AT performs the physical layer protocol which includes modulation/coding, demodulation/decoding, and mobile fading channel. Inside the IBM PC AT, there are two IBM synchronous data link control (SDLC) communication adapters which are connected to the protocol analyzers via RS232-C cables. This setup allows us to intercept all data communicating between two protocol analyzers.

Theoretically, three protocol analyzers should be used to represent three distinct network elements. However, in our scenario, all three elements do not communicate with each other simultaneously. For example, at the phase of making connection (call-setup) and relinquishment (call-clearing), either the calling party or the called party needs to communicate with the NMC, whereas, at the data transmission phase, only the calling party and the called party are active. Therefore, only two protocol analyzers are necessary to perform the protocol simulation. Figure 8 shows a command flow diagram among two protocol analyzers (emulating three entities) and the IBM PC AT.

Using this test-bed, we can simulate various scenarios in the mobile fading environment. Protocol parameters can be adjusted to optimize the system performance. It has been shown that this protocol is efficient to combat the disturbances, such as packet error, packet loss, and packet insertion, during connecting, data transmission, and relinquishment phases.

REFERENCES

1. G.H. Knouse and P.A. Castruccio, "The Concept of an Integrated Terrestrial/Land Mobile Satellite System," IEEE Trans. on Veh. Tech., Vol. VT-30, August 1981. pp.97-101.

2. R.E. Anderson, R.L.Frey, and J.R.Lewis, "Technical Feasibility of Satellite-Aided Land Mobile Radio," Proc. Inter. Conf. on Comm., June 1982. pp.7H2.1-7H2.5.

3. F. Naderi, G.H. Knouse, and W.J. Weber, "NASA's Mobile Satellite Communications Program; Ground and Space Segment Technologies," Proc. International Astronautics Federation Congress, Laussane, Switzerland, October, 1984.

4. W.J. Weber and F. Naderi, "NASA Mobile Satellite Experiment (MSAT-X)" Proc. National Electronic Conf., Vol. 37, Chicago, Illinois, June 1983. pp.328-330.

5. V.O.K. Li and T.-Y. Yan, "An Integrated Voice and Data Multiple Access Scheme for a Land Mobile Satellite System," Proc. of the IEEE, Special Issue on Satellite Communication Networks, Nov. 1984.

6. W. Stallings, Handbook of Computer Communications Standards, Volume 1, A Stallings/MacMillan Book, 1987.

7. A.S. Tanenbaum, Computer Networks, Prentice Hall, Inc., 1981.

8. C.C. Wang and T.-Y. Yan, "Performance Analysis of An Optimal File Transfer Protocol for Integrated Mobile Satellite Services", Proceeding of GLOBECOM'87, Tokyo, Japan, December, 1987.

9. R.J. Deasington, X.25 Explained: Protocols for Packet Switching Networks, Second Edition, Halsted Press, 1986.

FROM DTE TO NMC	FROM NMC TO DTE
CALL-REQUEST	INCOMING-CALL
CALL-ACCEPTED	CALL-CONNECTED
CLEAR-REQUEST	CLEAR-INDICATION
DTE-CLEAR-CONFIRMATION	DCE-CLEAR-CONFIRMATION

Figure 1. X.25 Control Packet Types for Connecting and Relinquishment Procedures

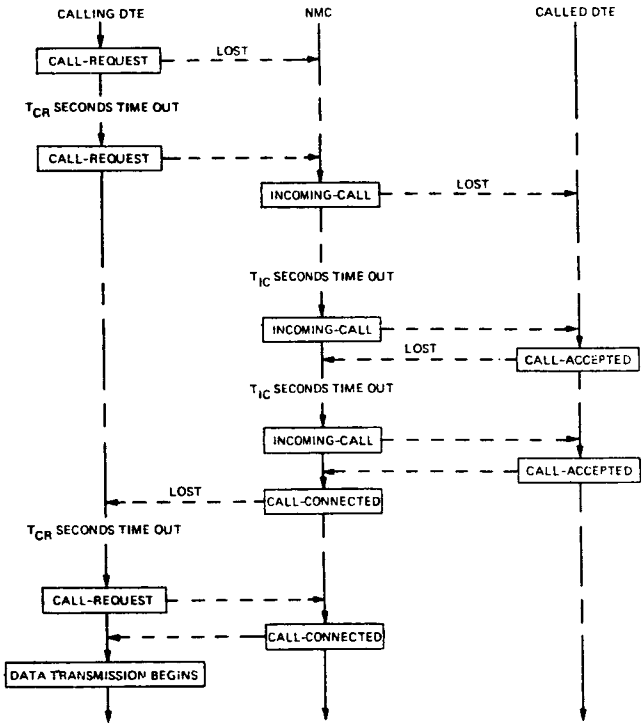

Figure 2(a). Establishing an Open-End File Transfer Session: Line-Not-Busy Case

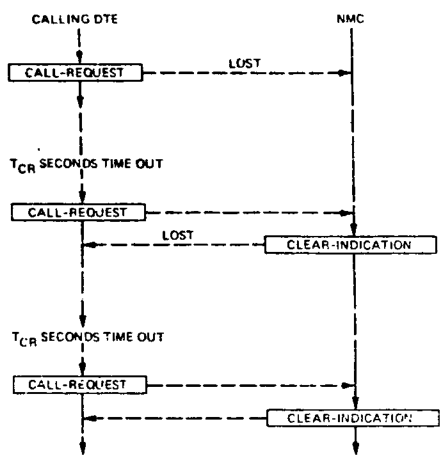

Figure 2(b). Establishing an Open-End File Transfer Session: Line-Busy Case

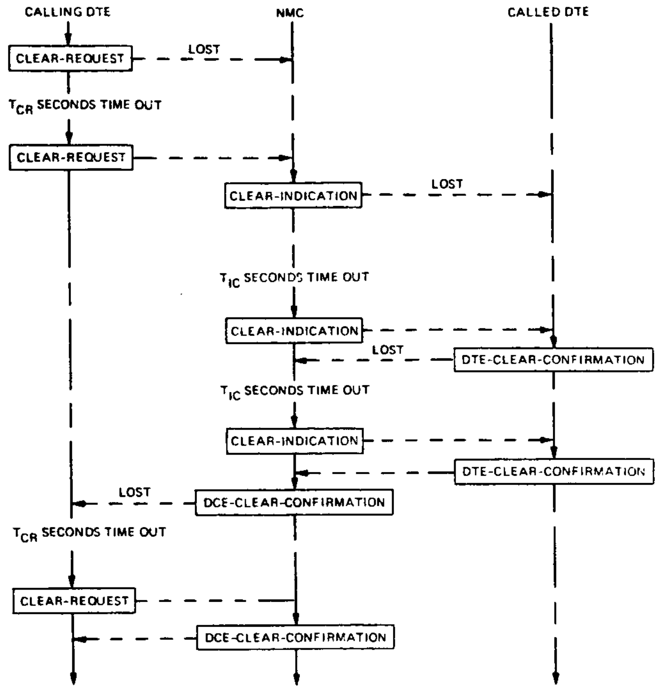

Figure 3. Relinquishing an Open-End File Transfer Session

BITS	8	7	6	5	4	3	2	1
OCTETS 1	0	0	0	1	GROUP #			
2	CHANNEL #							
3	0	0	0	0	1	0	1	1
4	CALLING ADDRESS LENGTH				CALLED ADDRESS LENGTH			
5	UP TO 14 OCTETS ADDRESSES							
n_1	0	0	FACILITY FIELD LENGTH					
n_2	OPTIONAL FACILITY FIELDS UP TO 16 OCTETS							

Figure 4(a). CALL-REQUEST and INCOMING-CALL Packet Format

BITS	8	7	6	5	4	3	2	1
OCTETS 1	0	0	0	1	GROUP #			
2	CHANNEL #							
3	0	0	0	0	1	1	1	1

Figure 4(b). CALL-ACCEPTED and CALL-CONNECTED Packet Format

BITS	8	7	6	5	4	3	2	1
OCTETS 1	0	0	0	1	GROUP #			
2	CHANNEL #							
3	0	0	0	1	0	0	1	1

Figure 4(c). CLEAR-REQUEST and CLEAR-INDICATION Packet Format

BITS	8	7	6	5	4	3	2	1
OCTETS 1	0	0	0	1	GROUP #			
2	CHANNEL #							
3	0	0	0	0	0	1	1	1

Figure 4(d). DCE-CLEAR-CONFIRMATION and DTE-CLEAR-CONFIRMATION Packet Format

FLAG	ADDR	CTRL	DATA	FCS	FLAG

FLAG : 01111110
ADDR : '00000011' COMMANDS FROM DCE TO DTE AND RESPONSES FROM DTE TO DCE
'00000001' COMMANDS FROM DTE TO DCE AND RESPONSES FROM DCE TO DTE
CTRL : INFORMATION
SUPERVISORY
UNNUMBERED
DATA : FROM NETWORK LAYER
FCS : FRAME CHECK SEQUENCE (2 BYTES)

Figure 5. Structure of X.25 Data Link Layer Frame

FORMAT	COMMANDS	BITS:	8	7	6	5	4	3	2	1
I	I			N(R)		P		N(S)		0
S	RR			N(R)		PF	0	0	0	1
S	RNR			N(R)		PF	0	1	0	1
S	REJ			N(R)		PF	1	0	0	1
*S	SREJ			N(R)		PF	1	1	0	1
U	SABM		0	0	1	P	1	1	1	1
U	UA		0	1	1	F	0	0	1	1
U	FRMR		1	0	0	F	0	1	1	1
U	DISC		0	1	0	P	0	0	1	1

* FOR HDLC ONLY

Figure 6. Control Field Format for Various X.25 Frames

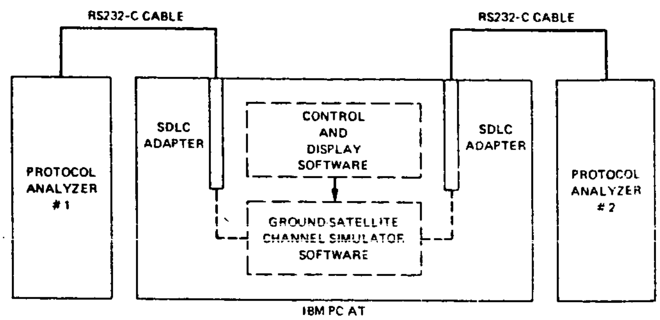

Figure 7. Hardware Configuration of a Test-Bed Simulator for Open-End File Transfer

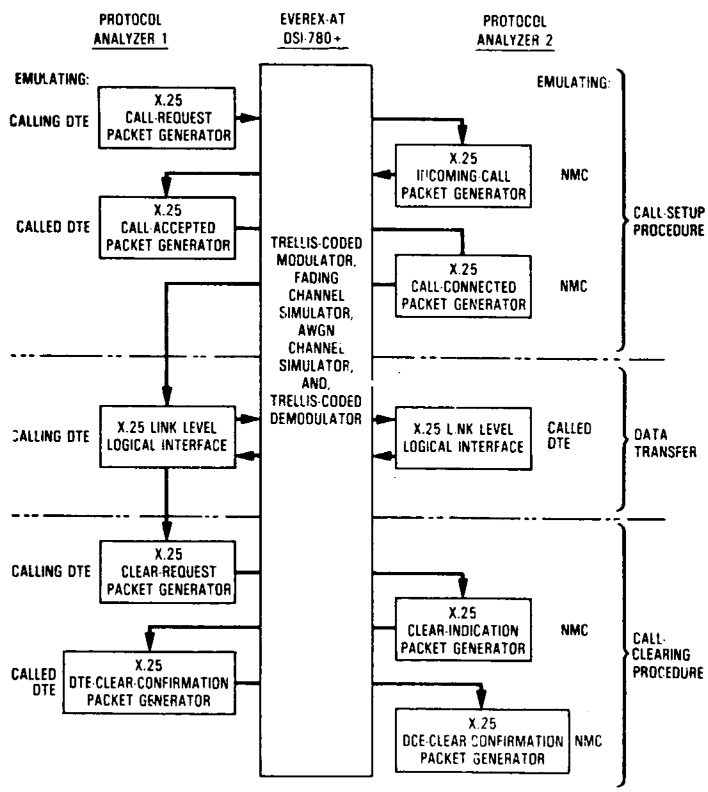

Figure 8. Control Flow Diagram of the Test-Bed Simulation